IPSO Smart Objects
and related IoT Standards

Jaime Jiménez jaime.jimenez@ericsson.com

12th February 2019

Table of Contents

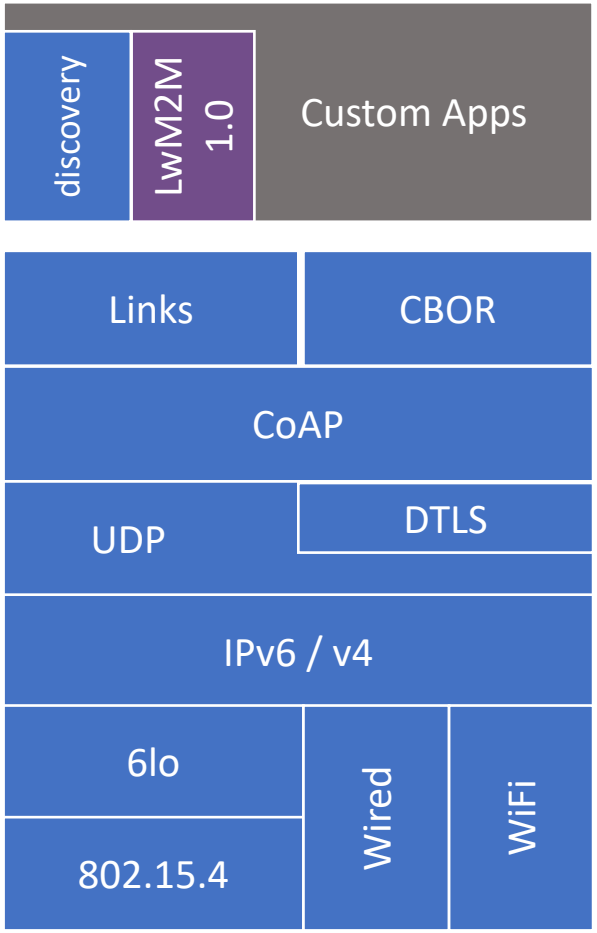
Internet Engineering Task Force (IETF)

IETF: dozen+ years of IoT standards

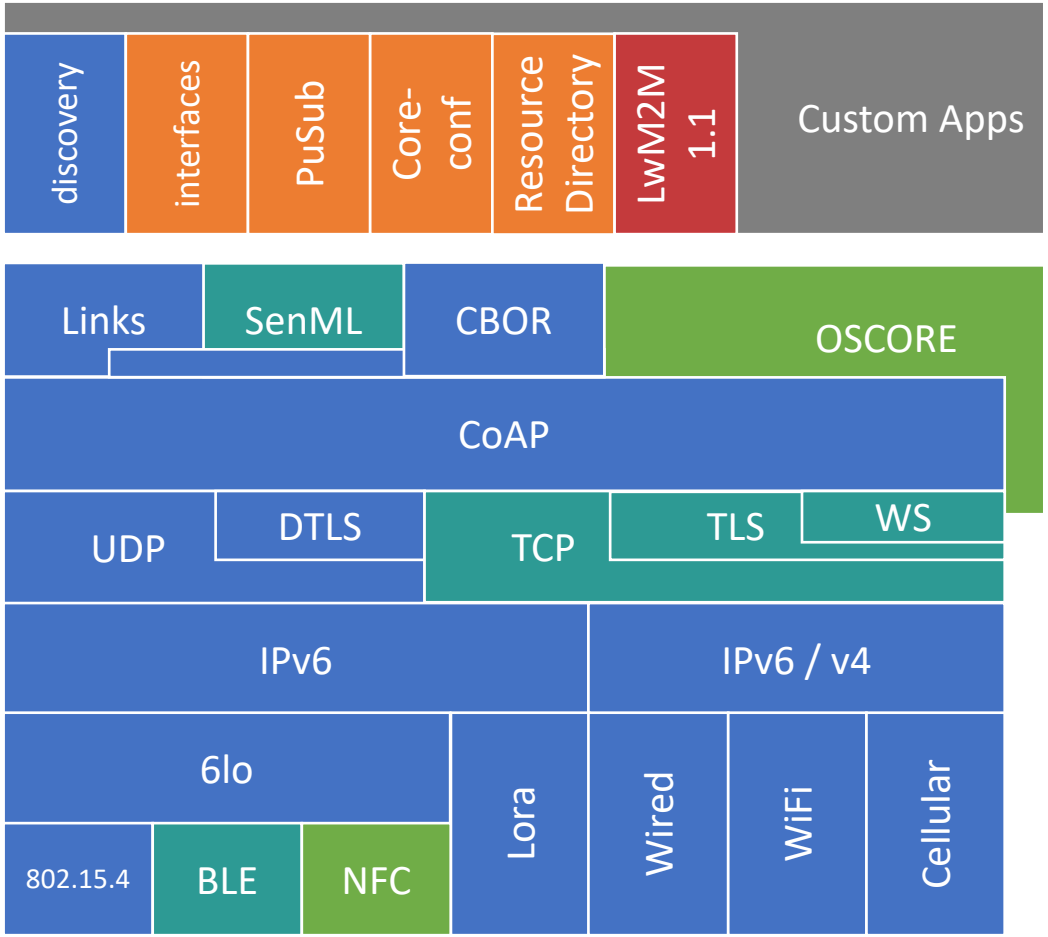
RFC 2689	RFC 3485	RFC 3544	RFC 3819	RFC 3940	RFC 3941	RFC 4629
RFC 4919	RFC 4944	RFC 5049	RFC 5401	RFC 5740	RFC 5856	RFC 5857
RFC 5858	RFC 6282	RFC 6469	RFC 6568	RFC 6606	RFC 6775	RFC 6690
RFC 7049	RFC 7228	RFC 7252	RFC 7388	RFC 7390	RFC 7400	RFC 7641
RFC 7668	RFC 7744	RFC 7925	RFC 7959	RFC 8075	RFC 8132	RFC 8152
RFC 8307	RFC 8323	RFC 8376	RFC 8392	RFC 8424	RFC 8516	…and more

Connectivity WGs Application WGs Security WGs

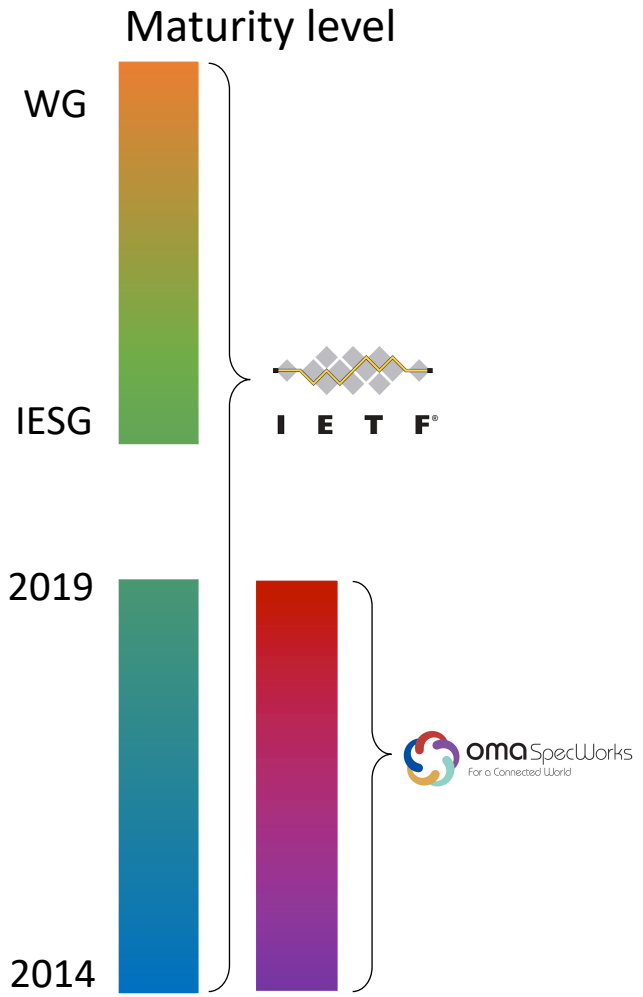
Standards Device Stack



2014



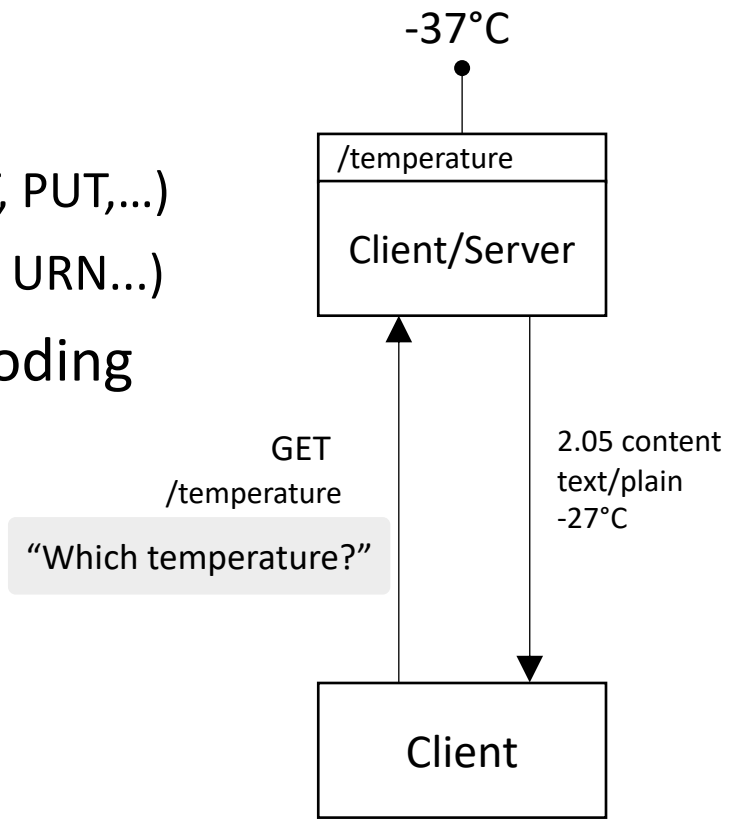
2019



Constrained Application Protocol (CoAP)

The Constrained Application Protocol (CoAP)

- CoAP (RFC7252) implements HTTP's **REST** model
  - Simple devices: 100 to 250 KiB code and 10 to 50 KiB RAM
  - Each device can be client and server exposing resources
  - CoAP defines methods to access those resources (GET, POST, PUT,…)
  - Same key concepts borrowed from HTTP (Media types, URL, URN...)
- Has a compact 4-byte header, with simple options encoding
- Simple protocol, datagram (UDP, DTLS)
  - Reliability through header message type "*CON/NON*"
  - With TCP/TLS (RFC8323) support for NAT-ed environments
- The Resource Directory provides a directory service

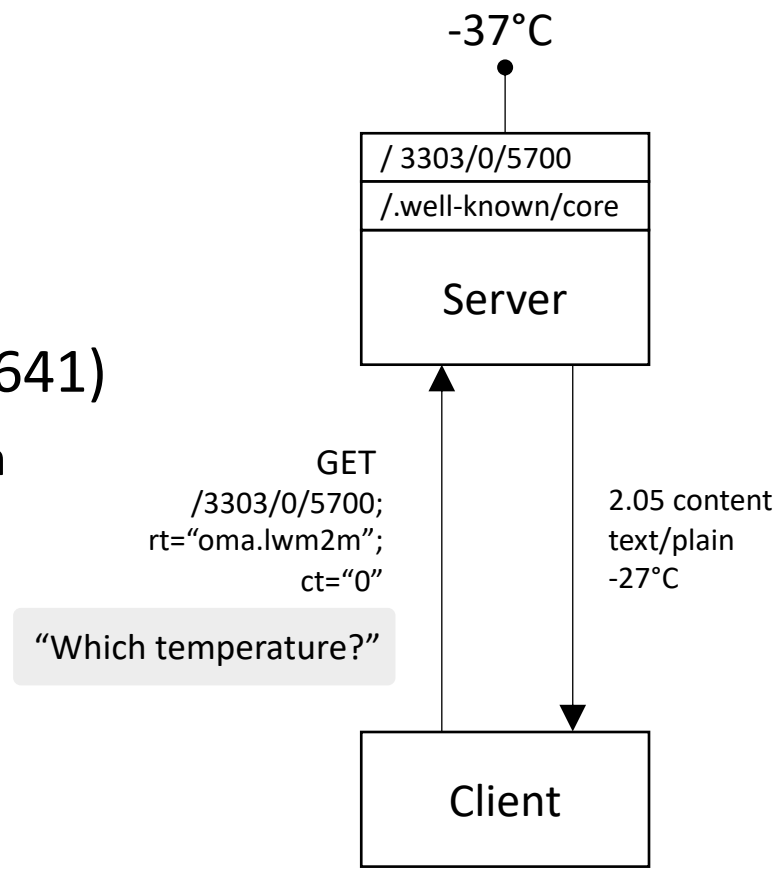


The Constrained Application Protocol (CoAP)

- CoRELink (RFC6690) provides a link format
  - Reuses Web Linking RFC5988 for IoT.
  - Enables query parameters for discovery (lt, gt…)
  - Enables attribute and relation types (rt, if, sz).

    `<3303/0/5700>;rt="oma:lwm2m:temp";ct="0"`
- Notifications available through *observe* option (RFC7641)
  - Can observe and add query parameters to the observation

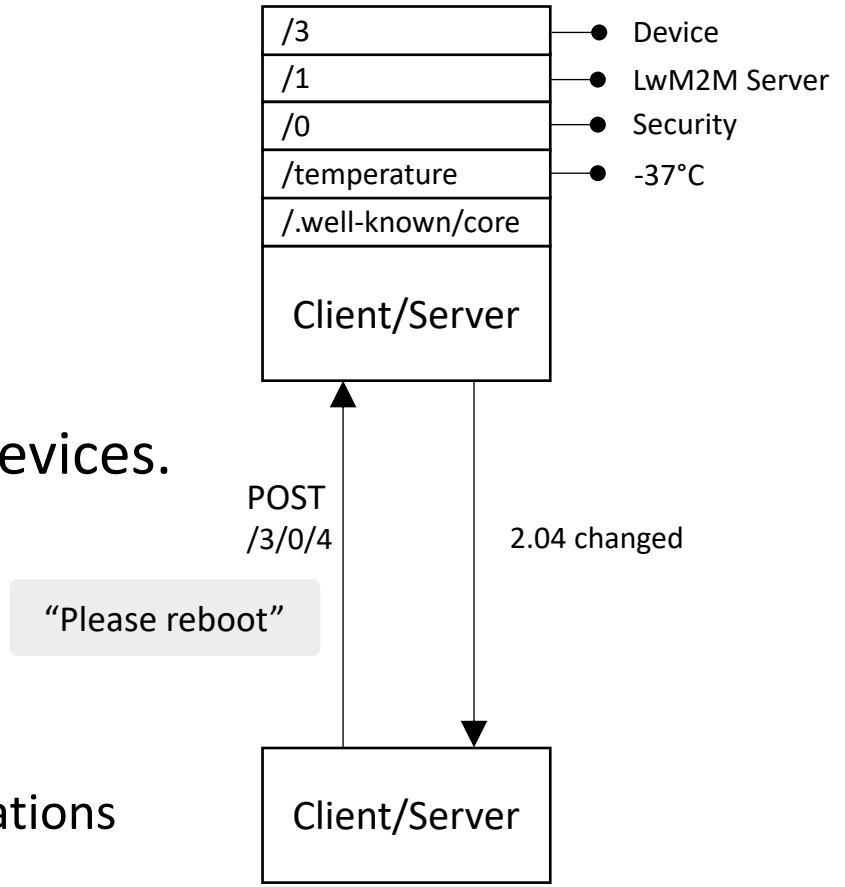
    `<3303/0/5700>?lt=0`
- The *"/.well-known/core"* URI provides discovery
- Multiple serialization formats used with CoAP
  - SenML (RFC8428): Minimalistic JSON
  - CBOR (RFC7049): Binary serialization
- Multiple implementations available at coap.technology



Lightweight M2M Protocol (LwM2M)

The LightWeight M2M Protocol (LwM2M)

- Used for REST type of Device Management
  - Runs on top of **CoAP,** on top of **IP**
  - Device = LwM2M Client = at least CoAP Server
  - LwM2M Server is the Manager entity
- Supports Resource Registration on LwM2M Server
- Provides a set of interfaces for managing of constrained devices.
  - Bootstrap: provisions device, configures keying
  - Client Registration: RFC6690 and RD
  - Information Reporting: enables event subscription
  - Device Management & Service Enablement: management operations

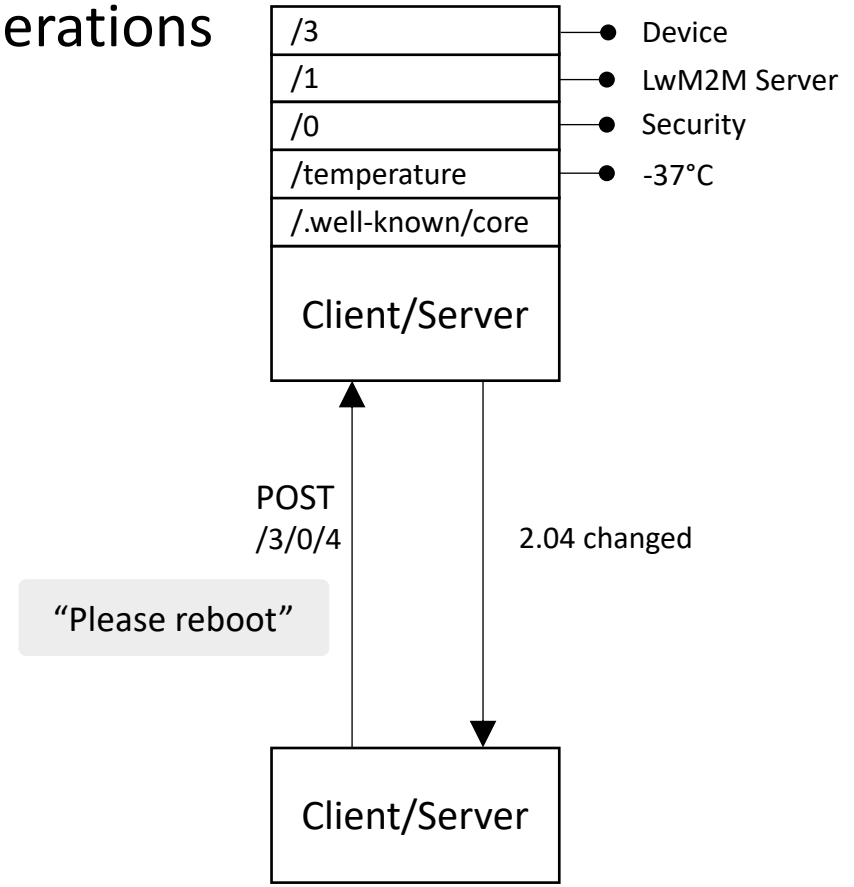


The LightWeight M2M Protocol (LwM2M)

- Mapping of CoAP methods (GET,POST, PUT…) to **CRUD** operations
- Interaction with device through simple “Objects”
  - RWX, Access Control, Observation, Notification
  - Independent from underlying protocol stack (CoAP today)
  - Simple resource structure
  - Objects’ resources are accessed with simple URIs:

    /{Object ID}/{Object Instance}/{Resource ID}
  - Multiple serializations:

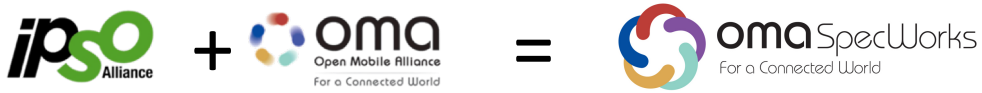
    For example JSON, CBOR and raw values.
- Common repository for all Objects (OMNA)
  - Enables interoperability and reusability



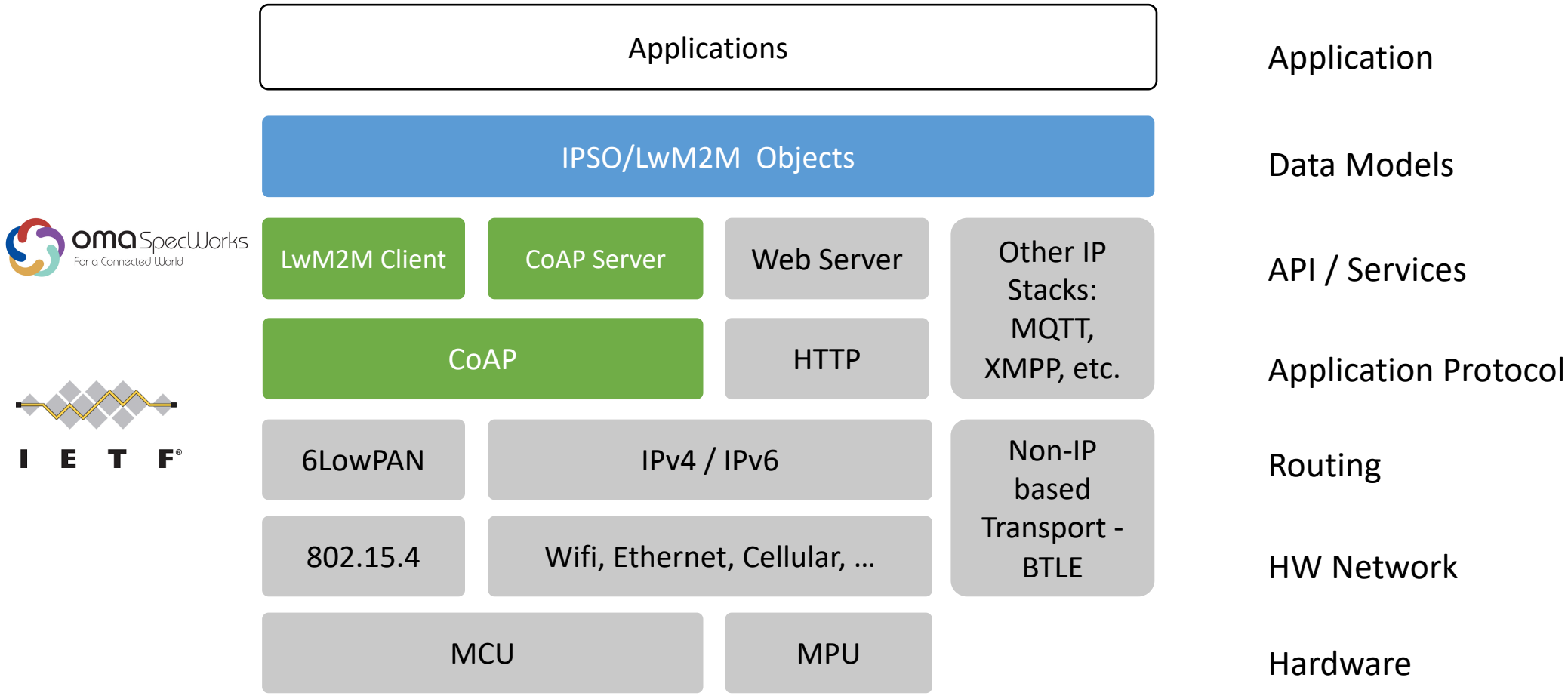
The LightWeight M2M Protocol (LwM2M)

Object Name	ID	Description
LwM2M Security	0	Keying material of a LwM2M Client to access a LwM2M server.
LwM2M Server	1	Data related to a LwM2M server.
Access Control	2	Information used to check whether a LwM2M Server has access to object.
Device	3	Device related information, including device reboot and factory reset function.
Connectivity Monitoring	4	Parameters related to network connectivity.
Firmware	5	Capability to update firmware
Location	6	Device location information
Connectivity Statistics	7	Information like transmit and receive counters
OSCORE	21	Provides security at the application layer

IPSO Smart Objects (IPSO)



The IP for Smart Objects (IPSO) device stack (recap)

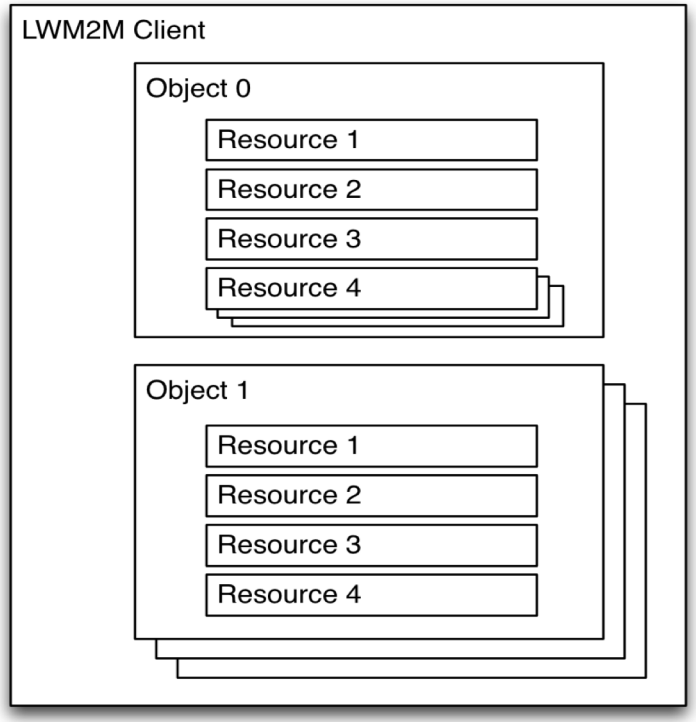


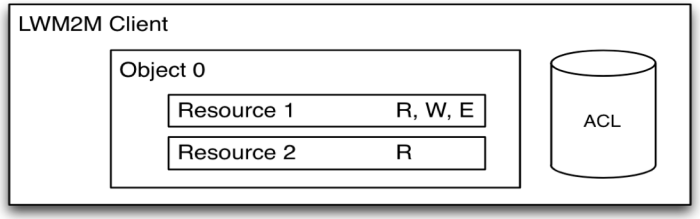
IPSO Object Structure

- Same URIs as LwM2M : `/{Object ID}/{Object Instance}/{Resource ID}`

```
/3300/0/5700
- 3300  Temperature Sensor
- 0     Instance 0 of a Temperature Sensor
- 5700  Resource having the current value
```

- Data Types (String, Integer, …) as LwM2M
- Operations (Read, Write, Create…) as LwM2M
- Object Linking and Core Link
  - Object Linking is used to refer to Objects within the device.
  - Allows composition without nasty large nested structures
  - Allows for complex objects (i.e. appliance made of several sensors)
  - CoRE Link enables query parameters: `rt="urn:oma:lwm2m:temp"`
- Extensible data model
  - Only few "Mandatory" Resources to enable interoperability
  - Use of versioning for model updates





IPSO Example Temperature Object

Object definition

Name	Object ID	Instances	Mandatory	Object URN
Temperature	3303	Multiple	Mandatory	urn:oma:lwm2m:3303

Resource definitions

ID	Name	Operations	Instances	Mandatory	Type	Units	Description	
5700	Sensor Value	R	Single	Mandatory	Float	…	…	Data
5601	Min Measured Value	R	Single	Optional	Float	…	…	
5602	Max Measured Value	R	Single	Optional	Float	…	…	
5603	Min Range Value	R	Single	Optional	Float	…	…	Metadata
5604	Max Range Value	R	Single	Optional	Float	…	…	
5701	Sensor Units	R	Single	Optional	String	…	…	
5605	Reset Min and Max	X	Single	Optional	Opaque	…	…	Actions

Example new Object – Company X Thermostat

Object info:

Object	Object ID	Object URN	Multiple Instances?	Description
Smart Thermostat	12300	urn:oma:lwm2m:x:12300	Yes	Smart Thermostat with multiple settings

Resource Info:

Resource Name	Resource ID	Access Type	Multiple Instances ?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float		Per Units resource	Temperature measurement
Units	5500	R,W	No	Mandatory	String	ucum:degF, ucum:degC		Units for 5700
Application Type	5750	R,W	No	Optional	String			Name, e.g. "Hall Thermostat"
Cooling	5200	R	No	Optional	Boolean			1=cooling
Heating	5201	R	No	Optional	Boolean			1=heating
Heat Source	5203	R	No	Optional	String	"Emergency", "Normal"		Indicates heat source
Fan Timer Active	5204	R,W	No	Optional	Boolean			1=running
Fan Timeout	5205	R,W	No	Optional	String		UTS	Time for fan to stop
Energy Save Mode	5206	R,W	No	Optional	Boolean			1= Energy Save mode
Away Mode	5207	R,W	No	Optional	Boolean			0=Home, 1=Away
Setpoint	5208	R	No	Optional	Float			Desired Temperature
HVAC Mode	5209	R,W	No	Optional	String	"Heat", "Cool", "Heat-Cool"		System Mode
High Setpoint	5210	R,W	No	Optional	Float			Highest desired temperature
Low Setpoint	5211	R,W	No	Optional	Float			Lowest desired temperature
High Away Setpoint	5212	R,W	No	Optional	Float			Highest away mode temperature
Low Away Setpoint	5213	R,W	No	Optional	Float			Lowest away mode temperature

IPSO Smart Objects

Object	Object ID
Digital Input	3200
Digital Output	3201
Analogue Input	3202
Analogue Output	3203
Generic Sensor	3300
Illuminance Sensor	3301
Presence sensor	3302
Temperature Sensor	3303
Humidity Sensor	3304
Power Measurement	3305
Actuation	3306
Set Point	3308
Load Control	3310
Light Control	3311
Power Control	3312
Accelerometer	3313
Magnetometer	3314
Barometer	3315
Voltage	3316

Object	Object ID
Current	3317
Frequency	3318
Depth	3319
Percentage	3320
Altitude	3321
Load	3322
Pressure	3323
Loudness	3324
Concentration	3325
Acidity	3326
Conductivity	3327
Power	3328
Power Factor	3329
Distance	3330
Energy	3331
Direction	3332
Time	3333

Object	Object ID
Gyrometer	3334
Color	3335
GPS Location	3336
Positioner	3337
Buzzer	3338
Audio Clip	3339
Timer	3340
Addressable Text Display	3341
On/Off Switch	3342
Dimmer	3343
Up/Down Control	3344
Multiple Axis Joystick	3345
Rate	3346
Push Button	3347
Multi-state Selector	3348
Bitmap	3349
Stopwatch	3350

IPSO Reusable Resources

Resource	Resource ID
Digital Input State	5500
Digital Input Counter	5501
Digital Input Polarity	5502
Digital Input Debounce	5503
Digital Input Edge Selection	5504
Digital Input Counter Reset	5505
Current Time	5506
Fractional Time	5507
Min X Value	5508
Max X Value	5509
Min Y Value	5510
Max Y Value	5511
Min Z Value	5512
Max Z Value	5513
Latitude	5514
Longitude	5515
Uncertainty	5516
Velocity	5517
Timestamp	5518
Min Limit	5519
Max Limit	5520
Delay Duration	5521

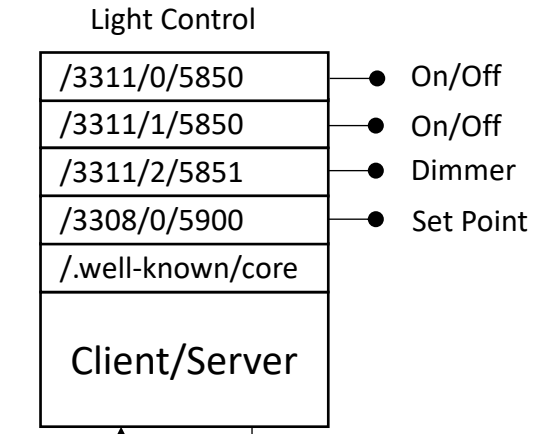
Resource	Resource ID
X Coordinate	5528
Y Coordinate	5529
Clear Display	5530
Contrast	5531
Increase Input State	5532
Decrease Input State	5533
Counter	5534
Current Position	5536
Transition Time	5537
Remaining Time	5538
Up Counter	5541
Down Counter	5542
Digital State	5543
Cumulative Time	5544
Max X Coordinate	5545
Max Y Coordinate	5546
Multi-state Input	5547
Level	5548
Digital Output State	5550
Digital Output Polarity	5551
Analog Input State	5600
Min Measured Value	5601

Resource	Resource ID
Reset Min and Max Measured Values	5605
Analog Output Current Value	5650
Sensor Value	5700
Sensor Units	5701
X Value	5702
Y Value	5703
Z Value	5704
Compass Direction	5705
Colour	5706
Application Type	5750
Sensor Type	5751
Instantaneous active power	5800
Min Measured active power	5801
Max Measured active power	5802
Cumulative active power	5805
Active Power Calibration	5806
Instantaneous reactive power	5810
Min Measured reactive power	5811
Max Measured reactive power	5812
Min Range reactive power	5813

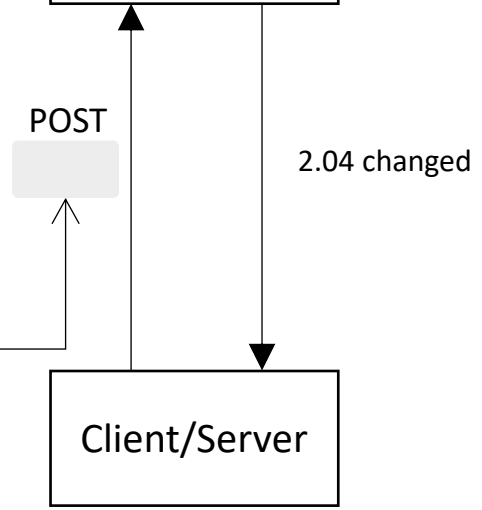
Resource	Resource ID
Reactive Power Calibration	5816
Power Factor	5820
Current Calibration	5821
Reset Cumulative energy	5822
Event Identifier	5823
Start Time	5824
Duration In Min	5825
Criticality Level	5826
Avg Load Adj Pct	5827
Duty Cycle	5828
On/Off	5850
Dimmer	5851
On Time	5852
Muti-state Output	5853
Off Time	5854
Set Point Value	5900
Busy to Clear delay	5903
Clear to Busy delay	5904
Bitmap Input	5910
Bitmap Input Reset	5911
Element Description	5912
UUID	5913

Example Actuation

- Actuation is handled with executionable (E) resources as well as readable and writeable ones (RW).
  - Parameters needed for actuation are passed along on resources.
  - Actuation can use executable and writeable (EW) resources.



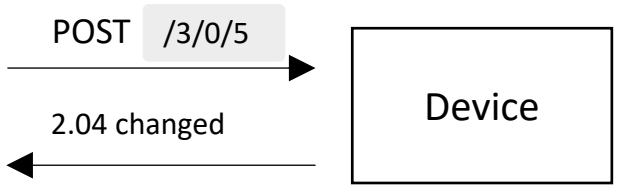
A server Write-Composite to switch off 2 light sources, dim a 3rd to 20% and set the thermostat to 18 degrees will have a JSON payload as shown in table below. Lights are all controlled by instances of IPSO Light Control Object (Object ID 3311), while thermostat is controlled by an instance of IPSO object Set Point (Object ID 3308).




```
[{"n":"/3311/0/5850", "vb":false},
{"n":"/3311/1/5850", "vb":false},
{"n":"/3311/2/5851", "v":20},
{"n":"/3308/0/5900", "v":18}]
```

Example Actuation

A command to instruct the device to perform a factory reset will have no payload and will be a simple POST to a specific resource. Factory Reset is part of LWM2M Device Object (Object ID 3).



A configuration command to use an industrial timer (Object ID 3340). The POST operation can be targeted to multiple resources like the trigger (Resource ID 5523), the duration of the time delay (Resource ID 5521) and the duration of the rearm delay after triggering (Resource ID 5525).

IPSO Serialization Formats

SenML - JSON

```
[{"bn":"/3/0/","n":"0","vs":"Open
Mobile Alliance"},
{"n":"1","vs":"Lightweight M2M
Client"},
{"n":"2","vs":"345000123"},
{"n":"3","vs":"1.0"},
{"n":"6/0","v":1},
{"n":"6/1","v":5},
{"n":"7/0","v":3800},
{"n":"7/1","v":5000},
{"n":"8/0","v":125},
{"n":"8/1","v":900},
{"n":"9","v":100},
{"n":"10","v":15},
{"n":"11/0","v":0},
{"n":"13","v":1367491215},
{"n":"14","vs":"+02:00"},
{"n":"16","vs":"U"}]
```

SenML-CBOR

```
90 a3 21 65 2f 33 2f 30 2f 00 61 30
03 74 4f 70 65 6e 20 4d 6f 62 69 6c
65 20 41 6c 6c 69 61 6e 63 65 a2 00
61 31 03 76 4c 69 67 68 74 77 65 69
67 68 74 20 4d 32 4d 20 43 6c 69 65
6e 74 a2 00 61 32 03 69 33 34 35 30
30 30 31 32 33 a2 00 61 33 03 63 31
2e 30 a2 00 63 36 2f 30 02 01 a2 00
63 36 2f 31 02 05 a2 00 63 37 2f 30
02 19 0e d8 a2 00 63 37 2f 31 02 19
13 88 a2 00 63 38 2f 30 02 18 7d a2
00 63 38 2f 31 02 19 03 84 a2 00 61
39 02 18 64 a2 00 62 31 30 02 0f a2
00 64 31 31 2f 30 02 00 a2 00 62 31
33 02 1a 51 82 42 8f a2 00 62 31 34
03 66 2b 30 32 3a 30 30 a2 00 62 31
36 03 61 55
```

SenML-CBOR diagnostic

```
[{-2: "/3/0/", 0: "0", 3: "Open
Mobile Alliance"}, {0: "1", 3:
"Lightweight M2M Client"},
{0: "2", 3: "345000123"},
{0: "3", 3: "1.0"},
{0: "6/0", 2: 1},
{0: "6/1", 2: 5},
{0: "7/0", 2: 3800},
{0: "7/1", 2: 5000},
{0: "8/0", 2: 125},
{0: "8/1", 2: 900},
{0: "9", 2: 100},
{0: "10", 2: 15},
{0: "11/0", 2: 0},
{0: "13", 2: 1367491215},
{0: "14", 3: "+02:00"},
{0: "16", 3: "U"}]
```

Implementations and OMNA Registry

- IPSO Objects are **MIT** license.
- Several Implementations support IPSO:
  - Example XML of the supported LwM2M and IPSO Objects in Leshan.
  - Sample C package for use of IPSO Objects in Contiki.
  - JS code templates of IPSO-defined devices code templates.
  - Sample Smart Objects Class can be used to create IPSO Smart Objects in your JavaScript applications.
  - BIPSO defines a set of BLE Characteristics that follows the IPSO Objects.
  - Contiki, Mbed and RIOT support IPSO Objects.
- Full object set available at the OMNA Registry:
  - http://www.openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html
- *"The total installed base of LwM2M-enabled devices will reach over 235 million units in 2022" (IDATE Digiworld)*